LETTER **Communicated by Terrence Sejnowski**

Mismatched Training and Test Distributions Can Outperform Matched Ones

Carlos R. González
crgonzal@caltech.edu
Yaser S. Abu-Mostafa
yaser@caltech.edu
Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, U.S.A.

In learning theory, the training and test sets are assumed to be drawn from the same probability distribution. This assumption is also followed in practical situations, where matching the training and test distributions is considered desirable. Contrary to conventional wisdom, we show that mismatched training and test distributions in supervised learning can in fact outperform matched distributions in terms of the bottom line, the out-of-sample performance, independent of the target function in question. This surprising result has theoretical and algorithmic ramifications that we discuss.

1 Introduction

A basic assumption in learning theory is that the training and test sets are drawn from the same probability distribution. Indeed, adjustments to the theory become necessary when there is a mismatch between training and test distributions. As we discuss, a significant body of work introduces techniques that transform mismatched training and test sets in order to create matched versions. However, the fact that the theory requires a matched distribution assumption to go through does not necessarily mean that matched distributions will lead to better performance, just that they lead to theoretically more predictable performance. The question of whether they do lead to better performance has not been addressed in the case of supervised learning, perhaps because of an intuitive expectation that the answer would be yes.

The result we report here is that, surprisingly, mismatched distributions can outperform matched distributions. Specifically, the expected out-of-sample performance in supervised learning can be better if the test set is drawn from a probability distribution that is different from the probability distribution from which the training data had been drawn, and vice versa. In the case of active learning, this would not be so surprising

Neural Computation **27, 365–387** (2015)
doi:10.1162/NECO_a_00697

since active learning algorithms deliberately alter the training distribution as more information is gathered about where the decision boundary of the target function is, for example. In our case of supervised learning, we deal with an unknown target function where the decision boundary can be anywhere. Nonetheless, we show that a mismatched distribution, unrelated to any decision boundary, can still outperform the matched distribution, a surprising fact that runs against the conventional wisdom in supervised learning. We first put our result in the context of previous matching work and then discuss the result from theoretical and empirical points of view.

In many practical situations, the assumption that the training and test sets are drawn from the same probability distribution does not hold. Examples where this mismatch has required corrections can be found in natural language processing (Jiang & Zhai, 2007), speech recognition (Blitzer, Dredze, & Pereira, 2007), and recommender systems, among others. The problem is referred to as data set shift and sometimes is subdivided into covariate shift and sample selection bias, as described in Quiñonero-Candela, Sugiyama, Schwaighofer, and Lawrence (2009). Various methods have been devised to correct this problem and is part of the ongoing work on domain adaptation and transfer learning. The numerous methods can be roughly divided into four types (Margolis, 2011).

The first type is referred to as instance weighting for covariate shift, in which weights are given to points in the training set, such that the two distributions become effectively matched. Some of these methods include discriminative approaches as in Bickel, Brückner, and Scheffer (2007, 2009); others make assumptions regarding the source of the bias and explicitly model a selection bias variable (Zadrozny, 2004); others try to match the two distributions in some reproducing kernel hilbert space as kernel mean matching (Huang, Smola, Gretton, Borgwardt, & Schölkopf, 2007); others estimate directly the weights by using criteria as the Kullback-Liebler divergence as in KLIEP (Sugiyama, Nakajima, Kashima, Von Buenau, & Kawanabe, 2008) or least squares deviation as in LSIF (Kanamori, Hido, & Sugiyama, 2009), among others. Additional approaches are given in Rosset, Zhu, Zou, and Hastie (2004); Cortes, Mohri, Riley, and Rostamizadeh (2008), and Ren, Shi, Fan, and Yu (2008). All of these methods rely on finding weights, which is not trivial as the actual distributions are not known; furthermore, the addition of weights reduces the effective sample size of the training set, hurting the out-of-sample performance (Shimodaira, 2000). Cross-validation is also an issue and is addressed in methods like importance weighting cross validation (Sugiyama et al., 2008). Learning bounds for the instance weighting setting are shown in Cortes, Mansour, and Mohri (2010) and Zhang, Zhang, and Ye (2012). Further theoretical results in a more general setting of learning from different domains are given in Ben-David et al. (2010).

The second type of methods uses self-labeling or cotraining techniques so that samples from the test set, which are unlabeled, are introduced in the training set in order to match the distributions; they are labeled using the labeled data. A final model is then reestimated with these new points. Some of these methods are described in Blum and Mitchell (1998), Leggetter and Woodland (1995), and Digalakis, Rtischev, and Neumeyer (1995). A third approach is to change the feature representation, so that features are selected, discarded, or transformed in an effort to make training and test distributions remain similar. This idea is explored in various methods, including Blitzer et al. (2007), Blitzer, McDonald, and Pereira (2006), Ben-David, Blitzer, Crammer, and Pereira (2007), and Pan, Kwok, and Yang (2008), among many others. Finally, cluster-based methods rely on the assumption that the decision boundaries have low density probabilities (Gao, Fan, Jiang, & Han, 2008), and hence try to label new data in regions that are underrepresented in the training set through clustering, as proposed in Blum (2001), and Ng, Jordan, and Weiss (2002). (For a more substantial review on these and other methods, refer to Margolis, 2011, and Sugiyama & Kawanabe, 2012.)

However, while great effort has been spent trying to match the training and test distributions, a thorough analysis of the need for matching has not been carried out. This letter shows that mismatched distributions can in fact outperform matched distributions. This is important not only from a theoretical point of view but also for practical reasons. The methods that have been proposed for matching the distributions not only increase the computational complexity of the learning algorithms but also may result in an effective sample size reduction due to the sampling or weighting mechanisms used for matching. Recognizing that the system may perform better under a scenario of mismatched distributions can influence the need for, and the extent of, matching techniques, as well as the quantitative objective of matching algorithms.

In our analysis, we show that a mismatched distribution can be better than a matched distribution in two directions:

- For a given training distribution P_R, the best test distribution P_S can be different from P_R.
- For a given test distribution P_S, the best training distribution P_R can be different from P_S.

The justifications for these two directions, as well as their implications, are quite different. In a practical setting, the test distribution is usually fixed, so the second direction reflects the practical learning problem about what to do with the training data if they are drawn from a different distribution from that of the test environment. One of the ramifications of this direction is the new notion of a dual distribution. This is a training distribution P_R that is optimal to use when the test distribution is P_S. A dual distribution serves

as a new target distribution for matching algorithms. Instead of matching the training distribution to the test distribution, it is matched to a dual of the test distribution for optimal performance. The dual distribution depends on only the test distribution and not on the particular target function of the problem.

The organization of this letter is as follows. Section 2 describes extensive simulations that give an empirical answer to the key questions and a discussion of those empirical results. The theoretical analysis follows in section 3, where analytical tools are used to show particular unmatched training and test distributions that lead to better out-of-sample performance in a general regression case. The notion of a dual distribution is discussed in section 4. Section 5 explains the difference of the results presented and the dual distribution concept with related ideas in active learning, followed by the conclusion in section 6.

2 Empirical Results

Consider the scenario where the data set R used for training by the learning algorithm is drawn from probability distribution P_R, while the data set S that the algorithm will be tested on is drawn from distribution P_S. We show here that the performance of the learning algorithm in terms of the out-of-sample error can be better when $P_S \neq P_R$, averaging over target functions and data set realizations. The empirical evidence, which is statistically significant, is based on an elaborate Monte Carlo simulation that involves various target functions and probability distributions. The details of that simulation follow, and the results are illustrated in Figures 1 and 3.

We consider a one-dimensional input space, $\mathcal{X} \in [-1, 1]$. There is no loss of generality by limiting our domain because in any practical situation, the data have a finite domain and can be rescaled to the desired interval. We run the learning algorithm for different target functions and different training and test distributions, and we average the out-of-sample error over a large number of data sets generated by those distributions and over target functions; then we compare the results for matched and mismatched distributions.

2.1 Simulation Setup.

2.1.1 Distributions. We use 31 different probability distributions to generate R and S: 1 uniform distribution $\mathcal{U}(-1, 1)$, 10 truncated gaussian distributions $\mathcal{N}^*(0, \sigma^2)$ where σ is increased in steps of 0.3, 10 truncated exponential distributions $\mathrm{Exp}^*(\tau)$ where τ is increased also in steps of 0.3, and 10 truncated mixture of gaussian Distributions such that $MG(\sigma) = \frac{1}{2}\left(\mathcal{N}^*(-0.5, \sigma^2) + \mathcal{N}^*(0.5, \sigma^2)\right)$, with σ increased in steps of 0.25. By truncating the distributions, we mean that if X has a truncated gaussian

distribution such that $X \sim \mathcal{N}^*(0, \sigma^2)$ and $\tilde{X}$ has a gaussian distribution with $\tilde{X} \sim \mathcal{N}(0, \sigma^2)$, then

$$P(X \leq x) = \begin{cases} 0 & x \leq -1 \\ \frac{1}{Z} P_N(X \leq x) & -1 \leq x \leq 1, \\ 1 & x \geq 1 \end{cases} \tag{2.1}$$

where $P_N(X \leq x) = P(\tilde{X} \leq x)$ and Z is a normalizing constant ($Z = P_N(-1 \leq \tilde{X} \leq 1)$). This applies as well to the truncated exponential and mixture of gaussian distributions.

2.1.2 Data Sets. For each pair of probability distributions, we carry out the simulation generating 1000 different target functions, running the learning algorithm, comparing the out-of-sample performance, and then averaging over 100 different data set realizations. That is, each point in Figures 1 and 3 is an average over 100,000 runs with the same pair of distributions but with different combinations of target functions and training and test sets. The sizes of the data sets are $N_R = 100$ and 300 and $N_S = 10,000$, where N_R and N_S are the number of points in the training and test sets R and S.

2.1.3 Target Functions. The target functions $f : [-1, 1] \to [-1, 1]$ were generated by taking the sign of a polynomial in the desired interval. The polynomials were formed by choosing at random one to five roots in the interval $[-1, 1]$. The learning algorithm minimized a squared loss function using a nonlinear transformation of the input space as features. The non-linear transformation used powers of the input variable up to the number of roots of the polynomial plus a sinusoidal feature, which allows the model to learn a function that is close, but not identical, to the target. This choice of target functions allows the decision boundaries to vary in both number and location in each realization. Hence, the results presented do not depend on a particular target function, so that the distributions cannot favor the regions around the boundaries, as these are changing in each realization. Notice there is no added stochastic noise so that the two classes could be perfectly separated with an appropriate hypothesis set.

Out-of-Sample Error. The expected out-of-sample error in this classification task is estimated using the test set generated according to each of the P_S with $N_S = 10,000$. It is computed as the misclassification 0-1 loss, that is,

$$\mathbb{E}_{x,R}[Eout(x, R, f)] = \mathbb{E}_{x,R}[I[f(x) \neq h(x)]], \tag{2.2}$$

where $\mathbb{E}_x[\cdot]$ denotes the expected value with respect to the distribution of random variable x, $I[a]$ denotes the indicator function of expression a, $x \sim P_S$, R is the training data set generated according to P_R, and h is the learned function.

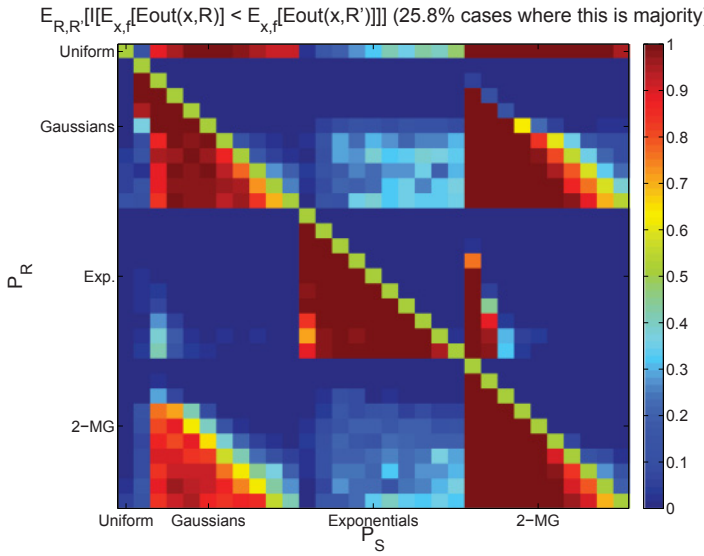

Figure 1: Summary of Monte Carlo simulation. Plot indicates, for each combination of probability distributions, $\mathbb{E}_{R\sim P_R, R'\sim P_S}[I[\mathbb{E}_{x\sim P_S, f}[Eout(x, R, f)] < \mathbb{E}_{f, x\sim P_S}[Eout(x, R', f)]]]$.

2.2 Fixing the Training Distribution. Figure 1 summarizes the result of the simulation to answer the question in the first direction. This result corresponds to the case where $N_R = 100$. Each entry in the matrix corresponds to a pair of distributions P_R and P_S. We fix P_R and evaluate the percentage of runs where using $P_S \neq P_R$ yields better out-of-sample performance than if $P_S = P_R$. That is, each entry corresponds to

$$\mathbb{E}_{R\sim P_R, R'\sim P_S}[I[\mathbb{E}_{f, x\sim P_S}[Eout(x, R, f)] < \mathbb{E}_{f, x\sim P_S}[Eout(x, R', f)]]]. \tag{2.3}$$

The matrix places families of distributions together, with increasing order of standard deviation or time constant. The result that immediately stands out is that in a significant number of entries, more than 50% of the runs have better performance when mismatched distributions are used, as indicated by the yellow, orange, and red regions, which constitute 25.8% of all combinations of the probability distributions used.

A number of interesting patterns are worth noting in this plot. The first row, which corresponds to $P_R = U(-1, 1)$, falls under the category of better performance for mismatched distributions for almost any other P_S used. There is also a block structure in the plot, which is no accident due to the way the families of distributions are grouped. Among these blocks, the

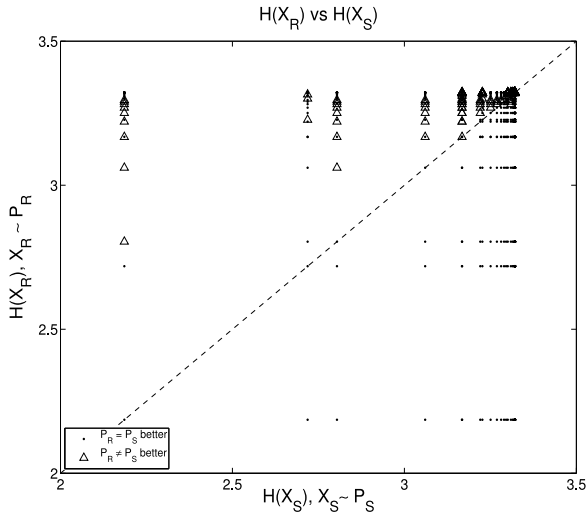

Figure 2: $H(X_R)$ versus $H(X_S)$: Characterization of why out-of-sample performance is better if there is a mismatch in distributions when P_R is fixed, using entropy.

lower triangular part of the blocks in the diagonal corresponds to cases where the distributions are mismatched but out-of-sample performance is better. We also note that the blocks in the upper-right and lower-left corners show the same pattern in the lower triangular part of the blocks.

Perhaps it is already clear to readers why this direction of our result is not particularly surprising, and in fact it is not all that significant in practice either. In the setup depicted in this part of the simulation, if we are able to choose a test distribution, then we might as well choose a distribution that concentrates on the region that the system learned best. Such regions are likely to correspond to areas where large concentrations of training data are available. This can be expressed in terms of lower-entropy test distributions, which are overconcentrated around the areas of higher density of training points. Such concentration results in a better average out-of-sample performance than that of $P_S = P_R$.

Figure 2 illustrates the entropy of different distributions. We plot $H(X_R)$ versus $H(X_S)$, where $H(\cdot)$ is the entropy and $X_R \sim P_S$ and $X_S \sim P_S$, marking the cases where using $P_S \neq P_R$ resulted in better out-of-sample performance of the algorithm. As it is clear from the plot, these cases occur when $H(X_S) < H(X_R)$.

A simple way to think of the problem is to see that if we could freely choose a test distribution and our learning algorithm outputs θ^* as the learned parameters that minimize some loss function $l(x, y, \theta)$ on a

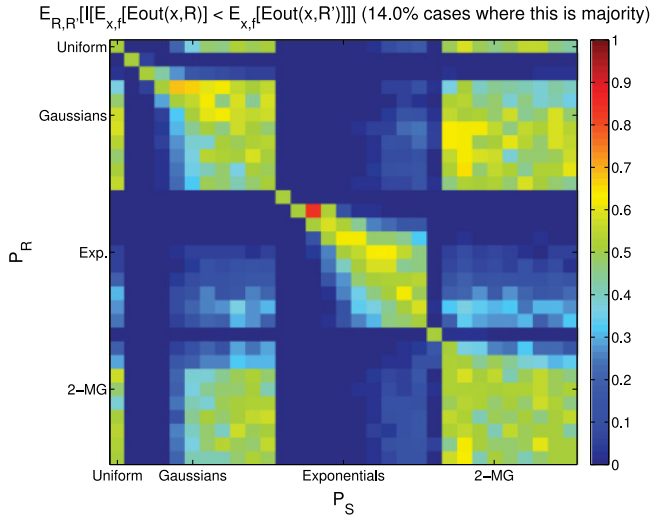

Figure 3: Summary of Monte Carlo simulation. The plot indicates, for each combination of probability distributions, $\mathbb{E}_{R\sim P_R, R'\sim P_S}[I[\mathbb{E}_{f,x\sim P_S}[Eout(x, R, f)] < \mathbb{E}_{f,x\sim P_S}[Eout(x, R', f)]]]$.

training data set $R = \{(x_i, y_i)\}$. Then to minimize the out-of-sample error, we would choose $P_S(x) = \delta(x - x^\star)$, where δ is the delta-dirac function and $x^\star = \arg\min_R(l(x, y, \theta^*)))$ the point in the input space where the minimum out-of-sample error occurs.

Results similar to those shown in Figure 1 are found when $N_R = 300$.

2.3 Fixing the Test Distribution. Figure 3 shows the result of the simulation in the other direction. Each entry in the matrix again corresponds to a pair of distributions P_R and P_S. However, this time we fix P_S and evaluate the percentage of runs where using $P_R \neq P_S$ yields better out-of-sample performance than if $P_R = P_S$. More precisely, once again, each entry computes the quantity in equation 2.3. Notice that this is the case that occurs in practice, where the distribution the system will be tested on is fixed by the problem statement. However, the training set might have been generated with a different distribution, and we would like to determine if training with a data set coming from P_S would have resulted in better out-of-sample performance. If the answer is yes, then one can consider the matching algorithms that we mentioned to transform the training set into what would have been generated using the alternate distribution that generated the training set.

The simulation result is quite surprising, as once again, there is a significant number of entries where more than 50% of the runs have better performance when mismatched distributions are used. For 14% of the entries, a mismatch between P_R and P_S results in lower out-of-sample error, as indicated by the light green, yellow, orange, and red entries in the matrix.

In this case, although the block structure is still present, there is no longer a clear pattern relating the entropies of the training and test distributions that allows explaining the result easily, as in the previous simulation. Notice that there are cases where the mismatch is better if we choose P_R of both lower and higher entropy than the given P_S. This is clear in the plot since the indicated regions in the block structure are no longer lower triangular but occupy both sides of the diagonal. This effect is analyzed further from a theoretical point of view in the following section. Since analyzing this effect theoretically is intractable in the case of classification tasks due to the nonlinearities, we carry out the analysis in a regression setting, noting that the Monte Carlo simulations show empirical evidence that the result also holds for the classification setting.

3 Theoretical Results

We now move to a theoretical approach to the above questions. We have shown empirical evidence that a mismatch in distributions can lead to better out-of-sample performance in the classification setting, and now we focus on the regression setting to cover the other major class of learning problems. In this section, we derive expressions for the expected out-of-sample error as a function of x, a general test point in the input space $\mathcal{X}$, and R, the training set, averaging over target functions and noise realizations. We will derive closed-form solutions as well as bounds that show the existence of $P_R \neq P_S$ with better out-of-sample performance than $P_R = P_S$.

We again consider the input space to be $\mathcal{X} = [-1, 1]$. We consider the usual regression setting where we are given a data set $R = \{(x_i, y_i)\}_{i=1}^N$, and we want to find the optimal parameter θ^* such that for a set of functions $\mathcal{H}$ parameterized by θ,

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - h(x_i; \theta))^2. \tag{3.1}$$

We let $\mathcal{H}$ to be the set of linear functions in some transformed space—that is,

$$h(x; \theta) = \theta^T \Phi_M(x), \tag{3.2}$$

where

$$\Phi_M(x) = [\phi_1(x)\ \phi_2(x)\ \cdots\ \phi_M(x)]^T \tag{3.3}$$

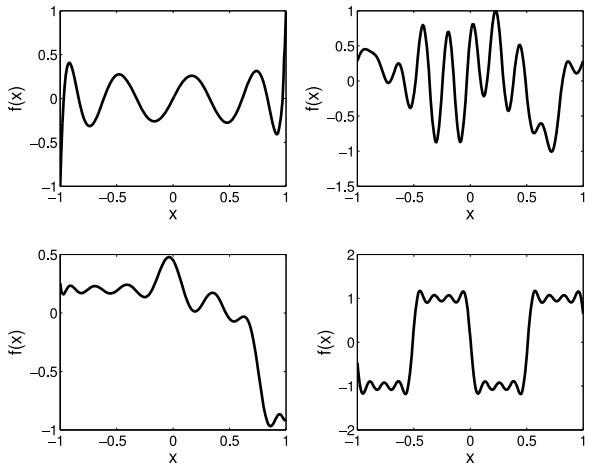

Figure 4: Sample realizations of targets generated with a truncated Fourier series of 10 harmonics.

is some nonlinear transformation of the input space defined by the set of basis functions $\{\phi_i\}_{i=1}^{M}$. For notational simplicity, let

$$z_M = \Phi_M(x). \tag{3.4}$$

We analyze the most general regression case, where there is both stochastic and deterministic noise (Abu-Mostafa, Magdon-Ismail, & Lin, 2012). We take $y_i = f(x_i) + \epsilon_i$, where ϵ_i represents the stochastic noise and f is more complex than the elements of $\mathcal{H}$, so $f \notin \mathcal{H}$, hence the deterministic noise.

We express the target function as

$$f(x) = \theta^T z, \tag{3.5}$$

where $z, \theta \in \mathbb{R}^C$, $z = \Phi_C(x)$ with $C \geq M$.

Using this formula for the target function allows for a wide variety of functions since C can be as large as desired, and we can use an arbitrary nonlinear transformation. Indeed, almost every function in the interval $\mathcal{X}$ can be expressed this way. For example, we could take the set of $\{\phi_i\}$ to be the harmonics of the Fourier series, so that with a large enough C, any function f that satisfies the Dirichlet conditions can be represented this way as a truncated Fourier series. Figure 4 shows just a few examples of the class of functions that can be represented using such a nonlinear transformation.

We reorganize the features in z and elements of θ as

$$z^T = \begin{bmatrix} z_M^T & z_C^T \end{bmatrix}, \quad \theta^T = \begin{bmatrix} \theta_M^T & \theta_C^T \end{bmatrix} \tag{3.6}$$

so that the first M features of z correspond to the features in the linear transformation that $\mathcal{H}$ can express. We further define Z and y as usual in regression problems,

$$Z = \begin{bmatrix} -z_1^T- \\ -z_2T- \\ \vdots \\ -z_N^T- \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \tag{3.7}$$

but now

$$Z = [Z_M \quad Z_C], \tag{3.8}$$

where $Z_M \in \mathbb{R}^{N\times M}$, $Z_C \in \mathbb{R}^{N\times(C-M)}$.

Finally, we make the usual independence assumption about the noise. Assume the stochastic noise has a diagonal covariance matrix $\mathbb{E}[\epsilon\epsilon^T] = \sigma_n^2 I$, where $\epsilon = [\epsilon_1\ \epsilon_2 \cdots \epsilon_N]^T$ and I is the identity matrix. Similarly, assume the energy of the features not included in $\mathcal{H}$ is finite, with $\mathbb{E}[\theta_C\theta_C^T] = \sigma_C^2 I$. For example, choosing Fourier harmonics as the nonlinear transformations guarantees a diagonal covariance matrix.

Consider now the out-of-sample error as a function of the point x in the input space, namely,

$$E_{out}(x, R) = (f(x) - h(x; \theta^*))^2, \tag{3.9}$$

where θ^* depends on the training set R and is given by the least-squares solution. Also notice that if we want to evaluate the out-of-sample error, then $x \sim P_S$,

$$\theta^* = Z_M^\dagger y = Z_M^\dagger(Z_M\theta_M + Z_C\theta_C + \epsilon), \tag{3.10}$$

and $Z_M^\dagger = (Z_M^T Z_M)^{-1} Z_M^T$.

Substituting, we get

$$\begin{aligned} \mathbb{E}_{\epsilon,\theta_C}[E_{out}(x, R)] &= \mathbb{E}_{\epsilon,\theta_C}\left[\left\|z^T\theta - z_M^T(Z_M^\dagger(Z_M\theta_M + Z_C\theta_C + \epsilon)\right\|^2\right] \\ &= \mathbb{E}_{\epsilon,\theta_C}\left[\left\|z_C^T\theta_C - z_M^T Z_M^\dagger(Z_C\theta_C + \epsilon)\right\|^2\right] \\ &= \sigma_C^2\left\|z_C^T - z_M^T Z_M^\dagger Z_C\right\|^2 + \sigma_N^2 z_M^T\left(Z_M^T Z_M\right)^{-1} z_M, \end{aligned} \tag{3.11}$$

where we have used the assumption about the noise and recall $z = \Phi(x)$ where $x \sim P_S$.

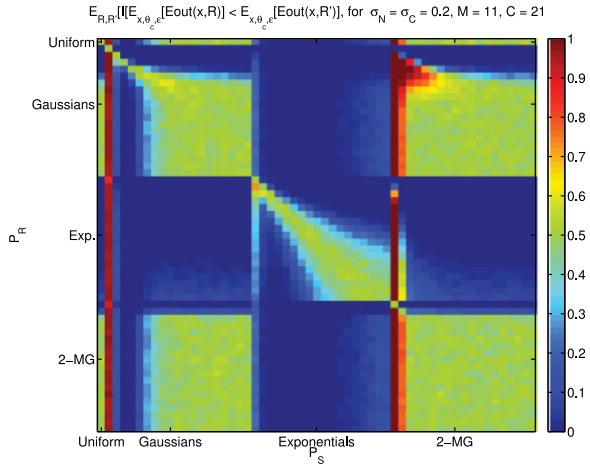

Figure 5: Monte Carlo simulation for $\mathbb{E}_{R\sim P_R, R'\sim P_S}[I[(\mathbb{E}_{x,\theta_C,\epsilon}[E_{out}(x,R)] < \mathbb{E}_{x,\theta_C,\epsilon}[E_{out}(x,R')]]]$, $M = 11, C = 21$, $N = 500$, and $\sigma_N = \sigma_C = 0.2$.

Notice that expression 3.11 is independent of θ as well as of the noise, and the only remaining randomness in the expression comes from generating R, which determines Z_M, and from z, the point chosen to test the error, making the analysis very general.

Now, we are interested in minimizing the expected out-of-sample error. Let R denote a training data set generated according to $P_R = P_S$ and R' a data set generated according to $P_R \neq P_S$. Can we find $P_R \neq P_S$ such that

$$\mathbb{E}_{R,x,\theta_C,\epsilon}[E_{out}(x,R)] > \mathbb{E}_{R',x,\theta_C}[E_{out}(x,R')]? \tag{3.12}$$

The simulation shown in section 2.3, although in a classification setting, suggests that this is the case. For completeness, we run the same Monte Carlo simulation in this regression setting. The advantage is that the closed-form expression found already averages over target functions and noise, allowing us to run in a shorter time more combinations of P_R and P_S, so that we only need to Monte Carlo the matrix Z. The expectation over $x \sim P_S$ can also be taken analytically with the closed-form expression found. In this case, we consider the same families of distributions, but we vary the standard deviation of the distribution in smaller steps to obtain a finer grid.

Figure 5 indicates that the question posed in equation 3.12 has an affirmative answer in 21% of the $P_R \neq P_S$ combinations that we considered. This particular simulation used the Fourier harmonics up to order 5, so that

$M = 11$, while the target used harmonics up to order 10, so that $C = 21$. Both $\sigma_C = \sigma_N = 0.2$, and $N = 500$. Each entry in the matrix computes

$$\mathbb{E}_{R\sim P_R, R'\sim P_S}[I[(\mathbb{E}_{x,\theta_C,\epsilon}[E_{out}(x,R)] < \mathbb{E}_{x,\theta_C,\epsilon}[E_{out}(x,R')]]], \tag{3.13}$$

which is the same quantity as that of equation 2.3, except that now f is determined by θ_C.

Notice that as shown in Figure 3, the cases where mismatched distributions outperform matched ones cannot be explained using an entropy argument, as was the case in section 2.2. Notice also that there are now combinations for P_R and P_S where almost 100% of the simulations returned lower out-of-sample error for mismatched distributions, especially when P_S was a truncated gaussian with small standard deviation ($\sigma = 0.2$) or when P_S was a mixture of two gaussians with $\sigma = 0.2$. In addition, we note the similarity between this simulation and the one shown for the classification setting in Figure 3.

We varied the size of N in order to see the effect of the sample size. We see very little variation in the results. Holding the other parameters constant, we obtain a very similar result. For $N = 1000$ and $N = 3000$, we obtain an affirmative answer to the question posed in equation 3.12 in 21% and 20% of the cases where $P_R \neq P_S$, respectively, so the result does not change from what we obtained in the $N = 500$ case. For $N = 100$, the percentage is even higher, at 30%. Hence, it is clear that although the number of combinations of distributions for which a mismatch between training and test distributions is larger for smaller N, the result still holds as N grows. Notice that in the simulations, the target function has 21 parameters. Hence, roughly for $N = 100$, there are effectively 5 samples per parameter, while for $N = 3000$, there are 150 samples per parameter. The latter is quite a large sample size given the complexity of the target function.

Going back to the derived expressions, a closed-form solution for the expected out-of-sample error is given by

$$\begin{aligned}\mathbb{E}[E_{out}(x,R)] = \mathbb{E}_R &\int_{-\infty}^{\infty} \sigma_C^2 \left\| z_C^T - z_M^T Z_M^\dagger Z_C \right\|^2 P_S(x)dx \\ &+ \int_{-\infty}^{\infty} \sigma_N^2 z_M^T \left(Z_M^T Z_M\right)^{-1} z_M P_S(x)dx.\end{aligned} \tag{3.14}$$

It cannot be further reduced analytically due to the inverse matrix terms. Yet, if we assume $C = M$ so that only stochastic noise is present, the expression reduces to

$$\begin{aligned}\mathbb{E}_{\epsilon,R,x}[E_{out}(x,R)] &= \mathbb{E}_R \int_{-\infty}^{\infty} \sigma_N^2 z^T (Z^T Z)^{-1} z P_S(x)dx \\ &\geq \sigma_N^2 \int_{-\infty}^{\infty} z^T (\mathbb{E}_R[Z^T Z])^{-1} z P_S(x)dx,\end{aligned} \tag{3.15}$$

where we use the result in Groves and Rothenberg (1969) for the expected value of the inverse of a matrix. With this expression, we can find an example of a mismatched training distribution that leads to better out-of-sample results. Again, without loss of generality, we pick the linear transformation consisting of Fourier harmonics, namely,

$$z = [1 \ \cos(\pi x) \sin(\pi x) \cdots \cos(m\pi x) \ \sin(m\pi x)]^T, \tag{3.16}$$

as this allows a vast representation of target functions. Here, $M = 2m + 1$.

If P_R is a uniform distribution over $\mathcal{X}$ or a gaussian distribution truncated to this interval, then

$$\begin{aligned} \mathbb{E}_R[Z^T Z] &= \mathbb{E}_R \sum_{i=1}^{N} z_i z_i^T \\ &= N\text{diag}(1, 0.5, 0.5, \ldots, 0.5). \end{aligned} \tag{3.17}$$

The above result is trivial for the uniform distribution case and can be easily evaluated with numerical integration for the truncated gaussians. This implies that

$$\begin{aligned} \mathbb{E}_{\epsilon,R,x}[E_{out}(x, R)] &\geq \sigma_n^2 \mathbb{E}_x \left[\frac{2m+1}{N} \right] \\ &= \frac{\sigma_n^2 M}{N}. \end{aligned} \tag{3.18}$$

Now instead, pick R' to be distributed according to Uniform$[-a, a]$. In this case,

$$\mathbb{E}_R[Z^T Z]_{ij} = \begin{cases} \text{sinc}(ja) & \text{if } i = 1,\ j \text{ is even} \\ \text{sinc}(ia) & \text{if } j = 1,\ i \text{ is even} \\ 1/2\,(1 + (-1)^i \,\text{sinc}(ia)) & \text{if } i = j \neq 1 \\ 1/2\,\big(\text{sinc}((i+j)a) & \text{if } i \neq j, \text{ and} \\ \quad + \text{sinc}((i-j)a\big) & i \text{ and } j \text{ odd} \\ 1/2(\text{sinc}\,\big((i+j)a) & \text{if } i \neq j, \text{ and} \\ \quad - \text{sinc}((i-j)a\big) & i \text{ and } j \text{ even} \\ 0 & \text{else} \end{cases}. \tag{3.19}$$

Figure 6 shows the closed-form bound for various choices of a and $M = 10$, choosing P_S to be a truncated gaussian with $\sigma = 0.4$. The dotted line

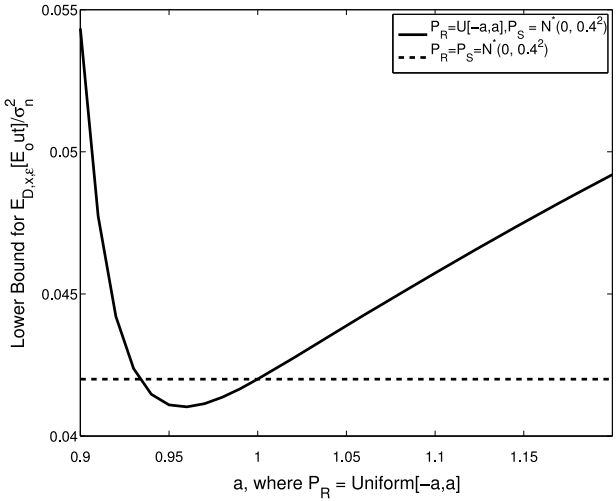

Figure 6: Bound for $\mathbb{E}_{R,x,\epsilon}[E_{out}(x,R)]$ when R is generated with $P_R = P_S = \mathcal{N}^*(0, 0.4^2)$ and for $P_R \neq P_S$ with $P_R = \text{Uniform}[-a, a]$.

shows the bound for the case $P_R = P_S$. As it is clear from the plot, there are various choices for a so that equation 3.12 is satisfied.

Since this is only a lower bound on the error, we verify that the minimum suggested by the bound does correspond to a superior mismatched distribution. We Monte-Carlo the value for both cases considered: we choose $P_S = \mathcal{N}^*(0, 0.4^2)$ and generate R according to P_S, while R' is generated according to $U[-0.97, 0.97]$. Notice that we use $a = 0.97$ because this choice results in the lowest error bound from Figure 6. Using $m = 10, N = 500$, and averaging over 10^8 realizations of R and R', we obtain

$$\mathbb{E}_{R,x,\theta,\epsilon}[E_{out}(x,R)] = 0.0440\sigma_N^2 > \mathbb{E}_{R',x,\theta,\epsilon}[E_{out}(x,R')] = 0.0429\sigma_N^2. \quad (3.20)$$

Hence, we have a concrete example of a distribution P_R that is different from P_S (see Figure 7) that leads to better out-of-sample performance, averaging over noise realizations and target functions. The existence of such distributions leads to the concept of a dual distribution, which we define in the following section.

4 Dual Distributions

Given a distribution P_S, we define a dual distribution $P_R^\star$ to be

$$P_R^\star = \arg\min_{P_R} \mathbb{E}_{R,x,f,\epsilon}[E_{out}(x,R)] \quad (4.1)$$

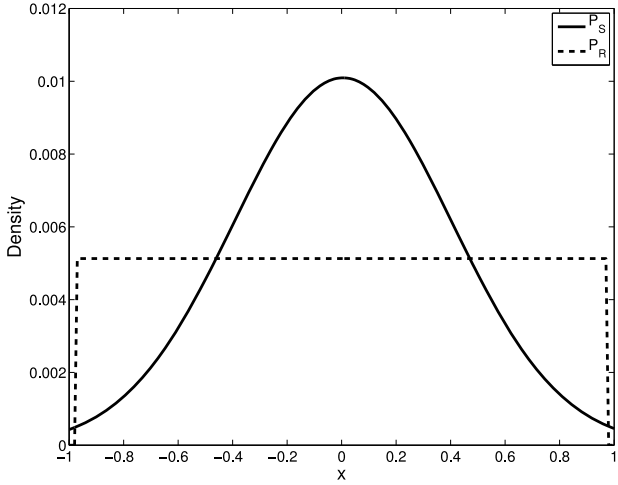

Figure 7: Pair of distributions $P_R \neq P_S$ such that expected out-of-sample error is lower when R is generated according to P_R rather than according to P_S for a regression problem in the domain $\mathcal{X} = [-1, 1]$.

where R is a data set generated according to P_R and $x \sim P_S$. As shown in the previous sections, it is not always the case that $P_R^\star = P_S$. The problem, of course, has the constraint that P_R must be a distribution.

We illustrate the concept of a dual distribution with an example where $P_R^\star$ can be readily found. Assume again that we want to solve a regression problem, but for simplicity, let us assume that only stochastic noise is present in the problem. Furthermore, we use a discrete input space $\mathcal{X} = \{x_i\}_{i=1}^d$, so that P_R and P_S are vectors, transforming the functional minimization problem into an optimization problem in $d - 1$ dimensions.

Given R, we can compute the expected out-of-sample error with respect to P_S, the noise, and the target functions as

$$\mathbb{E}_{x,\epsilon,\theta}[E_{out}(R)] = \sigma_N^2 \sum_{i=1}^{d} z_i^T (Z^T Z)^{-1} z_i P_S(x_i). \tag{4.2}$$

In this case, there are $\sum_{i=1}^{N} \binom{d}{i}$ possible data sets of size N (allowing for repetition of points in the data set) that could be obtained for any given P_R. To simplify the notation, since $\mathcal{X}$ is finite, we assign each of the points a number, from 1 to d, and we denote the out-of-sample error for each of these data sets as $E_{i_1,i_2,\ldots,i_N}$, where i_k indicates the element number in $\mathcal{X}$ that corresponds to the kth data point in R.

Hence, we can find the expected out-of-sample error with respect to P_R as

$$\mathbb{E}_{R,x,\epsilon,\theta}[E_{out}(R)] = \sum_{i_1,i_2,\ldots,i_N} p_{i_1} p_{i_2} \cdots p_{i_N} E_{i_1,i_2,\ldots,i_N}, \tag{4.3}$$

where all the $E_{i_1,\ldots,i_N}$ can be found with equation 4.2. Therefore, $P_R^\star$ is the solution to the following optimization problem:

$$\min_{p_1,p_2,\ldots,p_d} \sum_{i_1,i_2,\ldots,i_N} p_{i_1} p_{i_2} \cdots p_{i_N} E_{i_1,i_2,\ldots,i_N}, \tag{4.4}$$

$$\text{subject to } \sum_{i=1}^{d} p_i = 1$$

$$p_i \geq 0.$$

For illustration purposes, let $N = 3$:

$$\begin{aligned} z = \Phi(x) &= [\cos(\pi x)\ \sin(\pi x)]^T \\ \mathcal{X} &= \{-3/4, -1/4, 0, 1/4, 3/4\} \\ P_S &= [1/3, 0, 1/3, 1/3, 0], \\ [x_1, x_2, x_3, x_4, x_5] &= [-3/4, -1/4, 0, 1/4, 3/4]. \end{aligned} \tag{4.5}$$

Solving the optimization problem given in equation 4.4 yields $P_R^\star \neq P_S$, with

$$P_R^\star = [0.4672, 0.1140, 0.1140, 0.000, 0.3048]. \tag{4.6}$$

For this example,

$$\mathbb{E}_{R,x,\epsilon,\theta} = 1.5778\sigma_n^2 > \mathbb{E}_{R',x,\epsilon,\theta} = 1.1391\sigma_n^2, \tag{4.7}$$

where R is generated according to P_S and R' according to $P_R^\star$. Clearly there is a gain by training with the dual distribution. When running the optimization for data sets that have repeated points that result in undefined out-of-sample error, we conservatively take their error to be the maximum finite out-of-sample error over all combinations of possible data sets. Figure 8 shows the dual distribution found, along with the given P_S.

A very important property of the optimization problem, formulated in equation 4.4 is that it is a convex optimization program. In fact, it is a geometric program, although different from a standard geometric program since the equality constraint is not a monomial. Yet the problem is still convex. To illustrate this, let

$$\psi_i = \log(p_i), \tag{4.8}$$

$$L_{i_1,\ldots,i_N} = \log(E_{i_1,\ldots,i_N}). \tag{4.9}$$

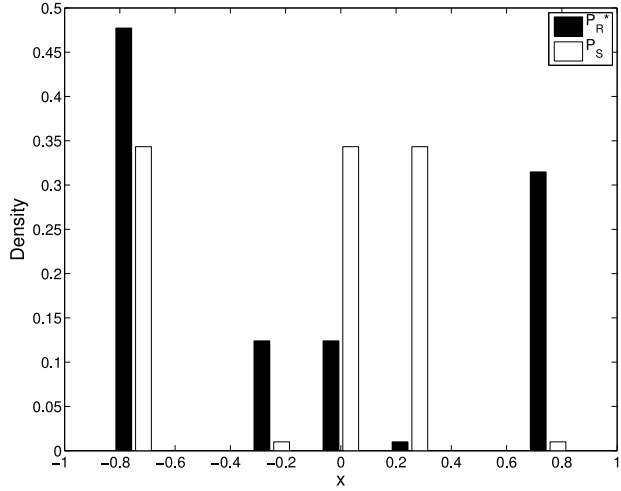

Figure 8: Probability mass functions for a given P_S and its dual $P_R^\star$ in a regression problem with stochastic noise, discrete input space $\mathcal{X} = \{-3/4, -1/4, 0, 1/4, 3/4\}$, and $N = 3$.

This change of variables implicitly makes $p_i > 0$ so that the inequality constraints can be removed. Also, the problem can be rewritten as

$$\min_{\psi_1,\psi_2,\ldots,\psi_d} \sum_{i_1,i_2,\ldots,i_N} e^{\sum_{k=1}^{N} \psi_{i_k} + L_{i_1,i_2,\ldots,i_N}}, \tag{4.10}$$

$$\text{subject to } \sum_{i=1}^{d} e^{\psi_i} = 1. \tag{4.11}$$

Notice that the objective function is a sum of exponential functions of affine functions of the ψ_i. Since exponential functions are convex, affine transformations of convex functions are also convex, and sums of convex functions result in a convex function, the objective is convex (Boyd & Vandenberghe, 2004). Following the same argument, the equality constraint is also convex, so that the optimization problem is a convex program.

Hence, if a minimum is found, this is the global optimum with a corresponding dual distribution. This problem can be solved with any convex optimization package. Furthermore, in most applications, P_S is unknown and is estimated by binning the data, obtaining a discrete version of P_S. Hence, this discrete formulation is appropriate to find dual distributions in such settings.

In the continuous case, the problem of finding the dual distribution can be written as

$$\min_p J(p) = \int_{x_N} \cdots \int_{x_1} \ell(x_1, \ldots, x_N) \prod_{i=1}^{N} p(x_i) dx_1 \ldots dx_N \tag{4.12}$$

$$\text{subject to} \quad \int p(x) dx = 1$$

$$p(x) \geq 0,$$

where

$$\ell(x_1, \ldots, x_N) = \int_x \mathbb{E}_{\epsilon, f}[E_{out}(x, x1, \ldots, x_N)] P_S(x) dx. \tag{4.13}$$

Notice that for clarity, we write $E_{out}(x, R)$ as $E_{out}(x, x_1, \ldots, x_N)$. Also notice that in the regression case considered in the previous section, there is a closed-form solution for $E_{out}(x_1, \ldots, x_N)$. For simplicity, in the case where there is only stochastic noise present, it is

$$\mathbb{E}_{\epsilon, f}[E_{out}(x_1, \ldots, x_N)] = \sigma_N^2 \Phi(x)^T \left(\sum_{i=1}^{N} \Phi(x_i) \Phi(x_i)^T \right)^{-1} \Phi(x). \tag{4.14}$$

Functional optimization methods like functional gradient descent can be used to solve the above problem.

The existence of a dual distribution has the direct implication that the algorithms mentioned in section 1 should be used to match P_R to $P_R^\star$ rather than to P_S. This applies even to cases where P_R is in fact equal to P_S, as it is conceivable that there will be gains if we now match to a dual distribution using $P_R^\star$ as the quantitative objective for the matching algorithms. Hence, this new concept applies to every learning scenario in the supervised learning batch setting, not only to scenarios where there is a mismatch between training and test distributions.

5 Difference with Active Learning

The concept of a dual distribution in supervised learning is somewhat related to similar ideas in active learning and experimental design. Especially, the methods of batch active learning, where a design distribution is found in order to minimize the error, seem to be solving a similar problem to the dual distribution. However, the fundamental difference is that active learning finds such optimal distribution given a particular target function. Hence, most methods rely on the information given by the target function in order to find a better training distribution. A common example is when distributions give more weight to points around the boundaries of

the target function. Yet the problem of finding the dual distribution is independent of the target function. The Monte Carlo simulations presented, as well as the bounds shown, average over different realizations of target functions.

For example, Kanamori and Shimodaira (2003) describe an algorithm to find an appropriate design distribution that will lower the out-of-sample error. In the algorithm proposed, a first parameter is estimated with s data points, and with this parameter, the optimal design distribution is found. Having a new design distribution, $T-s$ points are sampled from it, and a final parameter is then estimated. Notice, however, that the optimal design distribution is dependent on the target function. In the results we present, if a dual distribution is found given a particular test distribution, such distribution is optimal independent of the target function.

Other papers in the active learning community that focus on linear regression (e.g., Sugiyama, 2006) seem closely related to our work. For these, results apply to linear regression only and consider the out-of-sample error conditioned on a given training set. The nice property of the out-of-sample error in linear regression is that it is independent of the target function. This is the reason that even in the active learning setting, the dependence of the target function disappears and the mathematical analysis looks similar to the one we present. Yet although our analysis is done with linear regression and hence uses similar mathematical formulas, our approach is based on averaging over realizations of training sets and of target functions in the supervised learning scenario rather than in the cases addressed in Kanamori and Shimodaira (2003) and Sugiyama (2006). Furthermore, the problem of finding the dual distribution and the results presented can be applied to other learning algorithms besides linear regression for the classification and regression problems in the supervised learning setting.

Another difference that may stand out is the way the design distribution is used once it is found in the active learning papers, as opposed to how we propose to use the dual distribution here. In the active learning scenario, points are sampled from the design distribution, but in order to avoid obtaining a biased estimator, as shown in Shimodaira (2000), the loss function is weighted for these points with $w(x) = q(x)/p(x)$, following their notation, where $q(x)$ is the test distribution ($P_S(x)$) and $p(x)$ is the design distribution found. Notice that in the simulations presented in section 3, we do not reweight the points but instead explicitly allow a mismatch between P_S and P_R. Furthermore, in the supervised learning setting, where the training set is fixed and we are not allowed to sample new points, we propose that matching algorithms, as the ones described in section 1, be used to match the given training set to the dual distribution. In this case, the objective is to have weights $w(x) = P_R^\star(x)/P_S(x)$, so that the training set appears distributed as the dual distribution. These weights are actually inverse to those used in the active learning algorithms described. Although we are aware that the estimator computed in the linear regression setting will be biased when we use the dual distribution, we are concerned with

minimizing the out-of-sample error, which takes into account both bias and variance; hence, we may obtain a biased estimator but improve the mean-squared error performance as shown both analytically and through the simulation in section 3.

Furthermore, the results shown in Shimodaira (2000) hold only in the asymptotic case, and since we are dealing with the supervised learning scenario where only a finite training sample is available, the same assumptions are not valid. Thus, it is no longer optimal to use the mentioned weighting mechanism when N is not sufficiently large, as also shown in Shimodaira (2000). In the active learning setting, it is desirable that as more points are sampled, the proposed algorithms have performance guarantees. Hence, the algorithms are designed to satisfy conditions such as the consistency of the estimator and unbiasedness in the asymptotic case, which explains why the active learning algorithms use the above-mentioned weighting mechanism. In our setting, minimizing the out-of-sample performance with a fixed-size training set is our main objective, which is why the two approaches differ.

6 Conclusion

We have demonstrated through both empirical evidence and analytical bounds that in a learning scenario, in both classification and regression settings, using a distribution to generate the training data that is different from the distribution of the test data can lead to better out-of-sample performance, regardless of the target function considered. The empirical results show that this event is not rare, and the theoretical bounds allow us to find concrete cases where this occurs.

This introduces the idea of a dual distribution, namely, a distribution P_R different from a given P_S that leads to the minimum out-of-sample error. Finding this dual corresponds to solving a functional optimization problem, which can be reduced to a convex d-dimensional optimization problem if we consider a discrete input space.

The importance of this result is that the extensive literature that proposes methods to match training and test distributions in the cases where $P_R \neq P_S$ can be modified so that P_R is matched to a dual distribution of P_S. This means that those methods may work even in cases where $P_R = P_S$.

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012). *Learning from data*. N.p.: AMLBook.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1–2), 151–175.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In J. C. Platt, D. Koller, Y. Singer, &

S. T. Roweis (Eds.), *Advances in neural information processing systems, 19*. Cambridge, MA: MIT Press.

Bickel, S., Brückner, M., & Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning* (pp. 81–88). New York: ACM.

Bickel, S., Brückner, M., & Scheffer, T. (2009). Discriminative learning under covariate shift. *Journal of Machine Learning Research*, *10*, 2137–2155.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics.

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 120–128). Stroudsburg, PA: Association for Computational Linguistics.

Blum, A. (2001). Learning form labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with cotraining. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* (pp. 92–100). New York: ACM.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Cortes, C., Mansour, Y., & Mohri, M. (2010). Learning bounds for importance weighting. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems, 23* (pp. 442–450). Red Hook, NY: Curran.

Cortes, C., Mohri, M., Riley, M., & Rostamizadeh, A. (2008). Sample selection bias correction theory. In Y. Freund, L. Györfi, G. Turán, & T. Zeugmann (Eds.), *Algorithmic learning theory* (pp. 38–53). New York: Springer.

Digalakis, V., Rtischev, D., & Neumeyer, L. (1995). Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, *3*(5), 357–366.

Gao, J., Fan, W., Jiang, J., & Han, J. (2008). Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 283–291). New York: ACM.

Groves, T., & Rothenberg, T. (1969). A note on the expected value of an inverse matrix. *Biometrika*, *56*(3), 690–691.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K., & Schölkopf, B. (2007). Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, & T. Hofmann (Eds.), *Advances in neural information processing systems, 19*. Cambridge, MA: MIT Press.

Jiang, J., & Zhai, C. (2007). Instance weighting for domain adaptation in NLP. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics.

Kanamori, T., Hido, S., & Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, *10*, 1391–1445.

Kanamori, T., & Shimodaira, H. (2003). Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, *116*(1), 149–162.

Leggetter, C., & Woodland, P. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, *9*(2), 171–185.

Margolis, A. (2011). *A literature review of domain adaptation with unlabeled data* (Tech. Report). Seattle: University of Washington. http://ssli.ee.washington.edu/~amargoli/review_Mar23.pdf

Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 2* (pp. 849–856). Cambridge, MA: MIT Press.

Pan, S., Kwok, J., & Yang, Q. (2008). Transfer learning via dimensionality reduction. In *Proceedings of the 23rd National Conference on Artificial Intelligence* (vol. 2, pp. 677–682). Cambridge, MA: MIT Press.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. (2009). *Dataset shift in machine learning*. Cambridge, MA: MIT Press.

Ren, J., Shi, X., Fan, W., & Yu, P. (2008). Type-independent correction of sample selection bias via structural discovery and re-balancing. In *Proceedings of the Eighth SIAM International Conference on Data Mining, SDM* (pp. 565–576). Philadelphia: SIAM.

Rosset, S., Zhu, J., Zou, H., & Hastie, T. (2004). A method for inferring label sampling mechanisms in semi-supervised learning. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems, 17* (pp. 1161–1168). Cambridge, MA: MIT Press.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, *90*(2), 227–244.

Sugiyama, M. (2006). Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, *7*, 141–166.

Sugiyama, M., & Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. Cambridge, MA: MIT Press.

Sugiyama, M., Nakajima, S., Kashima, H., Von Buenau, P., & Kawanabe, M. (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems, 20* (pp. 1433–1440). Cambridge, MA: MIT Press.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*. New York: ACM.

Zhang, C., Zhang, L., & Ye, J. (2012). Generalization bounds for domain adaptation. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 25*. Red Hook, NY: Curran.

Received April 17, 2014; accepted August 19, 2014.